

Amendments to the Specification

Pages 1 and 31. Amend the title as follows:

PROGRAM EXECUTION METHOD USING AN OPTIMIZING JUST-IN-TIME COMPILER

Page 1, lines 7-8. Amend the paragraphs spanning these lines as follows:

A1
The present invention relates to a program execution method for dynamically compiling a program that is created using a programming language, such as Java™. (“Java” is a trademark of Sun Microsystems, Inc.)

Page 19, lines 28-29. Amend the paragraphs spanning these lines as follows:

A2
At step 112 (S112), the program execution apparatus 1 terminates the process that was using the interpreter. The process S10 thereafter terminates at step 118 (S118).

Page 20, lines 4-7. Amend the paragraphs spanning these lines as follows:

A3
At step 116 (S116), the program execution apparatus 1 executes the native code that the compiler generated. The process S10 thereafter terminates at step 118 (S118).

Fig. 3 is a flowchart showing the process at S14 according to the present invention. The process starts at step 140 (S140).

Page 21, line 20. Amend the paragraph spanning this line as follows:

A4
Fig. 4 is a flowchart showing the compiled code process (S20) for a transfer shown in Fig. 3. The process starts at step 200 (S200).

Page 22, lines 8-9. Amend the paragraph spanning these lines as follows:

A5

Fig. 5 is a flowchart showing the processing (S22) performed in Fig. 4 when another proposed transfer point is searched for. The processing starts at step 220 (S220).

Page 23, lines 11-12. Amend the paragraph spanning these lines as follows:

A6

Fig. 6 is a flowchart showing the processing (S26) used to move the transfer point outside the loop, without modifying the loop in Fig. 4. The processing starts at step 260 (S260).

Page 24, lines 1-2. Amend the paragraph spanning these lines as follows:

A7

At step 272 (S272), the program execution apparatus 1 repeats the processing for loop 2 up to S282 for each variable VI in a set $(V_r - V_d) \cap V_d$ of local variables that require recalculation.

Page 24, lines 16-18. Amend the paragraph spanning these lines as follows:

A8

At step 280 (S280), the program execution apparatus 1 determines whether the set $(V_e - V_d) \cap V_d$ is empty. If the set $(V_e - V_d) \cap V_d$ is empty, program control advances to S282. In the other case, program control is shifted to S300.

Page 24, lines 23-25. Amend the paragraph spanning these lines as follows:

A9

At step 286 (S286), the program execution apparatus 1 determines whether the set $(V_r - V_d) \cap V_d$ is empty. If the set $(V_r - V_d) \cap V_d$ is empty, program control advances to S300. In the other case, program control is shifted to S288.

Page 25, lines 15-19. Amend the paragraphs spanning these lines as follows:

A10
Fig. 7 is a flowchart showing the processing (S32) for using the loop modification shown in Fig. 4 to move the transfer point outside the loop. The processing starts at step 320 (S320).

As is shown in Fig. 7, at step 322 (S322) in S32, for each transfer point the program execution apparatus 1 repeats the processing for loop 1 up to S334-S234.

Page 26, lines 18-21. Amend the paragraphs spanning these lines as follows:

A11
At step 212 (S212), the program execution apparatus 1 generates native code for the method, and advances to the processing (S34 in Fig. 8) for generating transfer information. The process S20 thereafter terminates at step 214 (S214).

Fig. 8 is a flowchart showing the processing (S34) for generating transfer information in Fig. 4. The processing starts at step 340 (S340).

Page 26, lines 26-27. Amend the paragraphs spanning these lines as follows:

A12
At step 344 (S344), the program execution apparatus 1 repeats for each transfer point the processing performed up to S328-S348 for loop 1.